AI Chess Bot Project Timeline

Period 1: (Sep 3 - Sep 17, 2024)

- **Set Up Initial Repository**:
    - asic structure.
    - Defined the project goals and outlined the scope in the README file.
- **Implement Chessboard and Pieces**:
    - Developed a basic chessboard layout using a 2D array representation.
    - Added classes for each chess piece (e.g., Pawn, Rook, Knight, etc.) with basic movement capabilities.
    - Implemented basic validation for piece movement according to chess rules.

Period 2: (Sep 18 - Oct 1, 2024)

- **Understanding Chess Rules**:
    - Conducted research on chess rules, including piece movement, capturing, and special moves (e.g., castling, en passant).
    - Studied common strategies and tactics used in chess.
- **Learning Basic Python Syntax**:
    - Completed tutorials on Python fundamentals to enhance programming skills.
    - Practiced using functions, loops, and data structures relevant to game development.

Period 3: (Oct 2 - Oct 16, 2024)

- **Implement Command-Line Chess Interface**:
    - Designed a simple text-based user interface to allow users to input moves and view the chessboard state.
    - Implemented user input handling to validate and execute moves.
- **Develop Chess Bot with Minimax Algorithm**:
    - Researched and understood the Minimax algorithm and its application in game AI.
    - Implemented the Minimax algorithm for the chess bot, allowing it to evaluate potential moves and select the optimal one based on the current game state.

Period 4: (Oct 17 - Oct 30, 2024)

- **Introduction and Implementation of Minimax**:
    - Refined the Minimax algorithm, ensuring it could evaluate multiple moves ahead by implementing depth control.
    - Began implementing Alpha-Beta Pruning to optimize the Minimax algorithm, reducing the number of nodes evaluated in the search tree.
- **Documentation Updates**:

- Updated the README file to reflect the current status of the project, including features implemented and upcoming tasks.
- Added comments and documentation within the codebase to enhance readability and maintainability.

Period 5: (Oct 31 - Nov 13, 2024)

Implement Drag-and-Drop Functionality and User Interface Enhancements

- **Design and Develop Drag-and-Drop Chessboard:**
  - Integrated the chessboard with a graphical user interface using Pygame.
  - Implemented drag-and-drop functionality for moving chess pieces.
  - Ensured that piece selection and placement align with valid moves and chess rules.
  -

Period 6: (Nov 13 - Nov 27, 2024)

- **Exploration of AI Algorithms**
  - Explored combining MCTS with Minimax for enhanced AI performance.
  - Decided to focus on refining the Minimax evaluation function instead, improving the bot's decision-making.
- **Enhancing the Evaluation Function**
  - Updated evaluation function to account for piece safety, king safety, positional advantages, and piece activity.
  - Optimized AI's depth control for better performance and more challenging gameplay.

Period 7: (Nov 27 - Dec 15th, 2024)

- **Algorithm efficiency improvements**
  - Improved the speed at which the AI calculates the next move by improving the move ordering function with additional evaluations like rewarding moves capturing more valuable pieces as well as moves that move a piece away from danger
  - Move ordering is important as I am using Alpha beta pruning and by providing an optimal ordering, I maximize the number of pruned nodes.
  - I also explored implementing quiescence search which is a technique to overcome the "horizon effect," but this was taking significant computing time so I opted to not implement it in the end
- **Improved static board evaluation**
  - Lastly, I also added another heuristic to my evaluate board function which is to check how many pieces of each colour are defending a important squares

Self assessment of learning and Future work Items

- **Self assessment of learning**
    - I believe I learned a lot throughout the course of this project in terms of how to approach projects that involve concepts I am unfamiliar with
    - At the beginning of the project I was really concerned with making the bot work in any way shape or form without really understanding the in and outs of how the minimax algorithm worked
    - Thus my bot was not very strong at the start and I didnt know how to make it stronger
    - It was once I took a step back and really understood Minimax with alpha beta pruning that I was able to make significant progress on the bots functionality
    - I started off my watching YouYube videos about the general Minimax algorithm and then looking at different resources online (forums on Stack Overflow and Reddit as well as peer reviewed articles) on how existing chess algorithms implement minimax
    - It was though a combination of all this information that I was able to build my very on chess bot with a custom static evaluation and move ordering function
- **Future work Items**
    - On any project there will always be room from improvement and this is no exception
    - In terms of functionality, I still need to add piece promotion which is both complex in terms of evaluation as well as the UI aspect
    - In terms of the effectiveness of my bot, there is tons of room for improvement. In Fact, there is no such thing as a perfect chess bot as chess has about 10^120 possible positions!
    - I could continually improve my bot by adding new heuristics or increasing computing power by leveraging cloud resources to look more and more moves ahead!